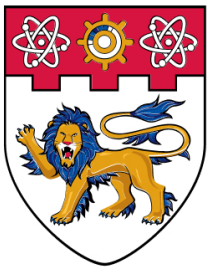

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4052 Cloud Computing

Assignment 2

Submitted by: Lim Zhi Qing

Matriculation Number: U2021897L

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

1. Introduction

This report will explore the design and implementation of a general quiz generator web application using Google GenAI. The backend for the quiz generator was deployed on Google Cloud Run. Text-bison was used as the language model for generating quiz questions. A simple proof-of-concept frontend was developed as the quiz generator web application.

2. Design and Implementation

```
PROMPT = """
Generate a quiz according to the following specifications:

- Topic: {topic}
- Number of Questions: {num_qns}
- Difficulty: {difficulty}
- Language: {language}

The output must be in the following format:
[
    {{
        "question": "question 1",
        "options": {{
            "a": "option a",
            "b": "option b",
            "c": "option c"
        }},
        "answer": "b"
    }},
    {{
        "question": "question 2",
        "options": {{
            "a": "option a",
            "b": "option b",
            "c": "option c"
        }},
        "answer": "a"
    }}
]

DO NOT ADD QUOTES OUTSIDE THE LIST
"""
```

Figure 1. Prompt Structure

```
[
  {
    "question": "Which of the following is not a type of cloud computing service model?",
    "options": {
      "a": "Infrastructure as a Service (IaaS)",
      "b": "Platform as a Service (PaaS)",
      "c": "Software as a Service (SaaS)"
    },
    "answer": "c"
  },
  {
    "question": "What is the term used to describe the process of moving data between a cloud storage service and
a local computer?",
    "options": {
      "a": "Data migration",
      "b": "Data synchronization",
      "c": "Data replication"
    },
    "answer": "a"
  }
]
```

Figure 2. Generated Questions

A Flask web application was implemented, using Vertex AI's Text Generation model. A predefined prompt was used, using user input to set the topic, number of questions, difficulty, and language of the quiz, as illustrated in Figure 1. A sample JSON format was given, and the prompt specifies that no quotes should be given outside the JSON data. An example of the generated questions in JSON format is illustrated in Figure 2.

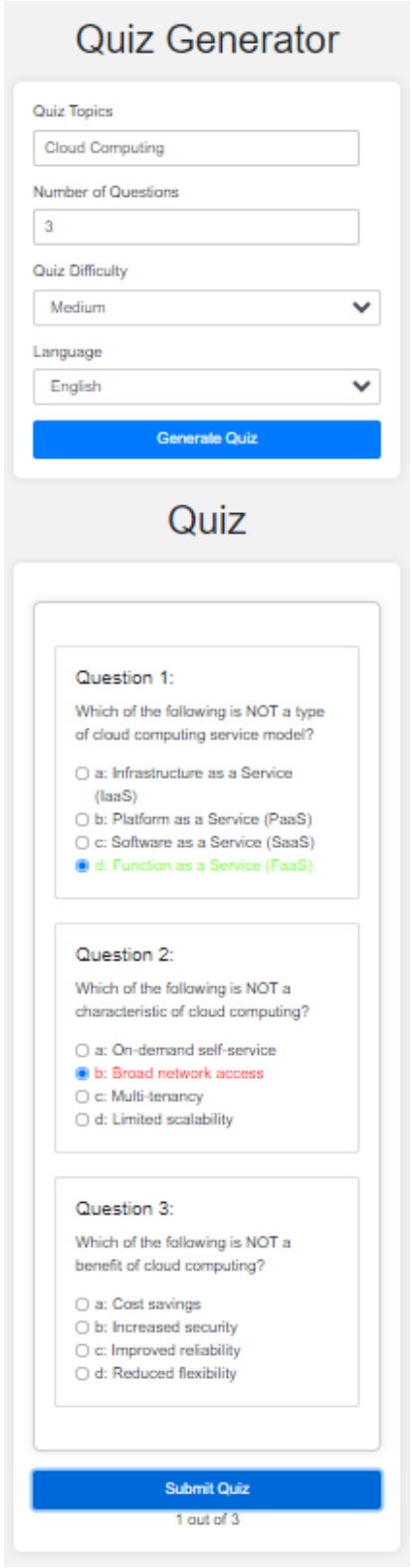

Figure 3. Quiz Generator Web Application

The Flask server was deployed on Google Cloud Run, a platform that allows containerised applications to be deployed and scaled automatically. A frontend application was developed to interact with the backend API. The web application is illustrated in Figure 3. The application contains 5 mandatory inputs. Quiz Topics specify the topics that are to be in the quiz. Number of questions must be in the range of 1 to 10. Quiz Difficulty can be set to easy, medium, or hard. Language can be set to English, Chinese, Malay, or Tamil. When the 'Generate Quiz' button is clicked, a request is sent to the Flask server with the specified inputs. A generated quiz will be returned and is displayed under the Quiz heading. Each question of the quiz presents options which can be selected. Once the user has finished selecting, the 'Submit Quiz' button can be clicked. Correct answers will be indicated in green. Incorrect answers will be indicated in red. Questions that are unanswered will be counted as incorrect. Finally, the score will be presented below the 'Submit Quiz' button. The user will be able to reselect their answers and resubmit.

The user input has a significant effect on the quiz generated. The quiz topics is fundamental to the content that is generated. However, quiz topics that are too long or too complex may not result in a good generated quiz. The number of questions has been limited to 10 or below as a value that is too high may exceed the number of tokens available for the text generation model. Quiz difficulty can also affect the questions generated, such as increasing or decreasing the number of options given. Finally, the language selection will render the entire quiz in the selected language.

3. Conclusion

The development and implementation of the quiz generator web application shows that cloud-based platforms such as Microsoft Azure and Amazon Web Services can be used in conjunction with large language models including BERT, LLaMA, GPT. The usefulness of these platforms and models allows developers to create dynamic applications that can leverage natural language processing capabilities to generate quizzes tailored to user preferences. This demonstrates the versatility and scalability of cloud computing environments as well as the potential of advanced language models in enhancing user experiences and facilitating innovative solutions across various domains, from education and entertainment to information retrieval and beyond.